High Speed Autonomous Embedded Software For High Accuracy Star Sensors

K.R.Muralidhara[#], Ashrita.M.S, Arpana A, S.Padmasree, J.Srinivas Rao, T.H.Shashikala, G.Nagendra Rao

Laboratory for Electro-Optics Systems (LEOS), ISRO,

Bangalore – 560058, India

[#]kr_muralidhara@leos.gov.in

***Abstract*— A star sensor is an embedded system used in modern spacecraft for estimating the orientation of the spacecraft. Star trackers capture image of star in the sky, process them and gave highly accurate attitude in all 3 axis. Basically a star tracker is an electronic camera connected to a microcomputer. First generation star trackers are characterized by acquiring and processing few stars information to estimate attitude with limited software modules. The availability of space qualified microcomputers enables generation shift in the star trackers development for attitude determination onboard. Laboratory for Electro-Optics Systems (LEOS) is actively involved in design and development of indigenous star trackers for different ISRO. Current star trackers employs traditional method of acquiring and processing star image and don't meet the many spacecraft requirements in terms of number of stars tracking, accuracy and update rate. In order to meet the current generation spacecraft requirements, LEOS has designed and developed second generation star trackers which are characterized by low weight, volume and power. These sensors are based on ERC-32 processor, here after referred as Mark-II sensor. The essential features expected from onboard software are robustness in operations, run-time and ability to give solutions especially during satellite injection with high angular rate. This paper presents the details of state-of-art software designed and developed for these star trackers to provide high accuracy and higher bandwidth with existing hardware limitations even with harsh space environment with limited resources and constraints.**

***Keywords*— Star sensor, Processor, Update rate**

I. INTRODUCTION

Modern 3-axis stabilized spacecraft uses star sensors to estimate the initial orientation of the spacecraft, star sensor is a satellite-based embedded system, which estimates the orientation of the satellite in space more accurately without prior information [1], and this information is essential for satellite control. Contemporary autonomous star trackers estimate the orientation directly from the image of stars taken by an onboard imaging instrument, which is part of star sensor. Basically, a star tracker is a electronic camera connected to a microcomputer. The camera part popularly called sensor head consists of camera control electronics and camera head electronics with baffle. The camera control electronics generates the signals needed to drive the camera head and formats the data for processing electronics, which is a microcomputer. The processed data is sent to AOCS through dedicated bus as shown in Fig.1.

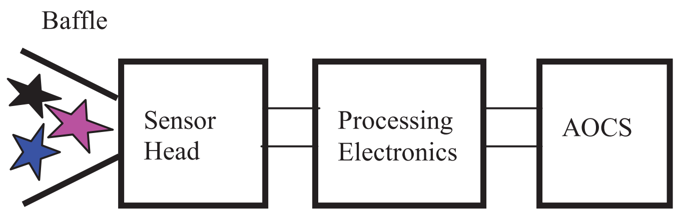

Fig.1: Block Diagram of Star Sensor.

As shown above, three systems can be identified in a star tracker, namely Optics, Sensing device and Microprocessor. The optics consists of systems of lenses that focus the light from a star on to a focal plane where the sensing device is located. Normally the sensing devices are Charge Coupled Device (CCD) or Active Pixel Sensors (APS). These detecting devices are normally based on array of light sensitive pixels that measure the intensity of the light they receive. These pixels are readout by a microprocessor that process their digitized output to estimate the position of the source of illumination within the star sensor Field of View (FOV). Conventional star sensors employed two basic modes for processing and estimating orientation of the spacecraft. When sensor is activated initially, it has no information about the satellite's orientation, this is known as the Lost-in-Space (LIS) or Initial Acquisition. Once the initial orientation is estimated this is used to estimate the orientation of the subsequent images, this is known as Tracking. It is based on predicting the current orientation and its rate of change accurately from previously obtained information. The main challenge is to provide fast LIS solution and more accurate and higher bandwidth in track mode.

LEOS of ISRO indigenously developed different types of star trackers. These star trackers are used in various remote sensing, communication, navigation and interplanetary and scientific missions. The first generation star trackers of LEOS are based on 16-bit processor like 8086 processor operates in only few traditional classic modes like acquisition and track and process only few stars with limited update rate due to many constraints. Second generation Mark-II star trackers are

978-1-4799-1441-8/13/$31.00 ©2013 IEEE

characterized by low weight, low volume and low power consumption with ERC-32 processor. Majority of the new generation spacecraft requirement like high accuracy, update rate and others are accomplished by state-of-art software designed and implemented specifically for these star trackers to drive sensor electronics and process the digitized data.

Many algorithms are used for processing and estimating the attitude in star trackers, the performance of these algorithms directly related to how these algorithms operate in different environment. Though these algorithms get the basic job done in batch and online environment, they fail to deliver many requirements in real time. The major requirement in such real time application is run-time or execution time, therefore software must strong enough to meet real time requirements. Though the first generation sensors are delivered basic requirements, but fail to deliver complete solutions because of hardware limitations. The run-time performance of software in real-time environment is based on the following factors.

1. Type of computing system
2. Detector type and method of image Processing.
3. Star database organization and Number of stars.
4. Languages used for implementing the algorithm.
5. Accessing mechanism of database.
6. Target processor used and its speed.
7. Memory architecture and Capacity.
8. Number of stars subjected for identification.
9. Minimum number of stars required for initial attitude.
10. Number of stars to be tracked in the FOV.
11. Criteria used to select valid stars for attitude.
12. Number of stars required for attitude determination.
13. Accuracy required and Update rate.

Improving the performance of the star sensor and to provide faster solution for efficient spacecraft controlling even with the above listed real time constraints is really a challenging task. We are presenting state-of-art star software modules designed, developed, implemented, tested to meet almost all new generation spacecraft requirements through massive software parallelism. The Mark-II star trackers with this software is successfully flown in missions like YOUTHSAT and SARAL and performance is excellent in all space conditions with different spacecraft operations like payload manoeuvres, thrusters firing and other operations.

II. Star Sensor Design

The Mark-II star tracker developed by LEOS is an autonomous star tracker. Like other state-of-art star trackers it has camera electronics and processing electronics. The camera head consists of imaging optics to collect star light from sky. To meet low weight and optimized optical performance, a seven element optics weighing about 350g indigenously developed in LEOS is used. A radiation hardened area array CCD of size 1024X1024 is used as detector. The processing electronics of sensor consists of ERC32 SPARC processor working at 12 MHz speed, in addition to main processor a custom made Video Processor (VIP) is used to perform the CCD related operations, this Video Processor acts as a co-processor for the main microprocessor. The processing Electronics consists of 3 different types of memories - PROM for Boot program, EEPROM as secondary storage and RAM as main memory. In addition to these memories, VIP has its own storage to deposit acquired and processed digitized data of star image. Once the data is deposited in the shared memory of VIP, the main processor fetches these data and performs the specified operations. The important specifications of the sensor are shown in Table-1.

TABLE-1

SUMMARY OF SENSOR SPECIFICATIONS

Sl No	Parameter	Specification
1	Field of View	20^0 X 20^0
2	Detector Size	1024 X 1024
3	Processor	ERC32
4	Processor Speed	10 MHz
5	Minimum Stars for Acquisition	5
6	Maximum Stars for Identification	17
7	Acquisition Time	1 sec
8	Maximum Tracked Stars	10
9	Minimum Stars for Track Mode	4
10	Update Rate	8 Hz
11	Cross Axis Accuracy	10”
12	Bore Sight Accuracy	40”
13	Angular Rate	1.5^0/sec
14	Angular Acceleration	$0.05^0/s^2$

III. Modes of Star Sensor

Majority of star sensors employed two basic modes for estimating orientation of the spacecraft namely Acquisition and Track mode and performs limited operations. LEOS star tracker has 15 different modes of operations to cater to the various requirements, which are broadly classified into two groups namely Operating modes and Service modes and these modes are listed below. Though the first three modes are used in onboard for estimating the attitude, the remaining modes are not least important. The remaining modes are used to increase the performance and good health of sensor at various stages. In this paper we focus only on the onboard operating modes, the remaining modes are out of scope of this paper.

1. Autonomous Acquisition Mode
2. Autonomous Track Mode
3. Autonomous Pointing Mode
4. Characterization Acquisition Mode
5. Characterization Track Mode
6. Calibration Mode
7. Self Test Mode
8. Memory Read Mode
9. Memory Write Mode
10. Memory Copy Mode
11. CRC Mode
12. EEPROM Copy Mode
13. Memory Compare Mode
14. Maintenance Mode and
15. StandBy Mode

IV. AUTONOMOUS POINTING MODE

The star tracker hardware functions in all imaging modes are more or less same and employ the standard philosophy, but the way we process the pixels data from detector makes star sensor operations in different modes. The optics is designed to focus the light from a point-like source, such as a star in a star image of circular shape and of a diameter equivalent to 2-3 times the side of a pixel. The pixels in detector are usually integrating devices and should ideally return the number of photons deposited upon them by the light source over the time interval called integration or exposure time. In reality, their operation is affected by noise. A pixel's output contains both a signal component which is proportional to the number of photons from the light source and a noise component. Once the integration is over the data is transferred from image zone of detector to memory zone. Based on the way in which memory zone of detector is processed, majority of star sensors employed two basic modes for estimating orientation of the spacecraft, they are acquisition mode and track mode as shown in Fig.2..

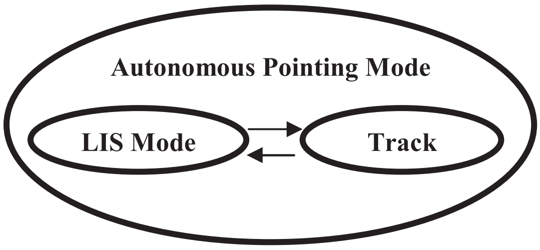

Fig.2: Transition in Autonomous Pointing Mode.

V. AUTONOMOUS ACQUISITION MODE

When sensor is activated initially, it has no information about the satellite's orientation, this is known as the Lost-in-Space (LIS) or Initial Acquisition, the Acquisition mode can be separated into three main parts.

1. Star centre estimation.
2. Star identification and
3. Attitude estimation.

In conventional star trackers, the above steps are carried out by processing complete array of detector with normal thresholding to detect star position. A rough estimation of centre of star position called Centroid is estimated which is used for further identification and attitude determination, all these activities are time consuming and fast LIS solution may not possible. Providing fast LIS solution for star sensors with the above time consuming activities and detection of star through illuminated pixels with a noise in LIS computation is highly challenging task. Therefore the performance of the LIS algorithm mainly depends on run-time performance of above operations and methodology adopted to process the pixels to detect star position. In order to provide faster LIS solution LEOS Mark-II star tracker software uses novel methods to perform all these operations efficiently and comparable to any state-of-art star trackers, which are explained below.

A. Star Centre Estimation

Detection of star is a highly challenging task, especially when sensor has a high noise level relative to the signal level makes detection difficult, because the illuminated pixels do not 'stand out'. To estimate star centre in LIS mode, a method called binning is used, in binned mode instead of processing pixels by pixels a group of adjacent pixels defined by bin factor is combined as a single pixel. The processing of the binned pixel outputs is carried out in two stages, first stage is a detection stage, where the binned pixels which have been illuminated by a star called “litpixels” are identified using defined threshold with Sobel operator. Secondly, using a cluster of contiguous pixels those have been marked as "litpixels", the exact position of the source of the illumination is estimated. This is done by assuming that contiguously illuminated pixels have been illuminated by a single star whose image on the pixel array is circular. The estimated star centroid is converted to direction cosines co-ordinates called measured co-ordinates.

B. Star Identification

The process of star identification is to associate body-frame measured star image directions with the catalog reference inertial directions. LEOS first generation star trackers used in many remote sensing satellites uses Pyramid star identification algorithm presented by Mortari [1], here after we refer this as Algorithm-A. The Algorithm-A has many disadvantages in real time space environment, mainly the worst case run time of Algorithm-A is high. This high run time forces Algorithm-A has to work in compromised environment, this leads to lower identification success rate even at normal spacecraft rate and completely fail to provide solution during high rotation rate of spacecraft in space environment. A state-of-art star identification algorithm here after this algorithm is referred as Algorithm-B is designed and developed for Mark-II star tracker. The run-time performance of this algorithm is comparable to state-of-art algorithm and the run-time performance is better by 4-5 times comparing to Algorithm-A. The success rate of Algorithm-B, is close to 100%, means provides identification solution at all times, which is major requirement of inter planetary and scientific missions. The state-of-art identification algorithm provides successful identification solution even at higher rotation rate of spacecraft. The successful star identification provides reference coordinates for the selected measured star vectors.

C. Attitude Estimation

Attitude estimation requires 2 set of vectors namely, measured vector and reference vector. As already explained measured vectors are out come centroid estimation process and reference vectors are out come of identification process. In this star tracker attitude is estimated in form of quaternions, two algorithms namely QUEST and Second Optimal Estimator of Quaternion ESOQ2 are studied and finally ESOQ2 is implemented. This estimates attitude much faster than QUEST, which detects the need of rotation much earlier stage at the computation of the attitude profile matrix itself.

VI. AUTONOMOUS TRACK MODE

Once the initial attitude is estimated, this is used to estimate the orientation of the subsequent images, this is known as Tracking. It is based on predicting the current orientation and its rate of change accurately from previously obtained information. In track mode unlike in LIS instead of processing complete array of detector only selected area of detector is processed by defining a window called track window, this is necessitated to improve the throughput of star tracker. The track window pixels are processed to find the centre of the star image applying a centroiding algorithm [3] which essentially takes a weighted average of the centers of the illuminated pixels where the weights are a function of the output of each illuminated pixel. The accuracy of star tracker mainly depends on accuracy of star Centroid, the centroiding of star can only be reliably performed if the noise component in the pixel outputs is not large. A high noise level relative to the signal level degrades the accuracy of the star centre. Under static conditions where star positions constant in the sensor's Field Of View (FOV) good signal-to-noise properties in the pixel outputs can be ensured by increasing the exposure time thus increasing the number of photons deposited on each illuminated pixel. When the star moves in the sensor's FOV, its image on the pixel array is also moving. Under such conditions, the time it takes the star image to sweep over a pixel puts an upper bound on the number of photons that can be collected by that pixel. The robustness of a star sensor to angular rates is therefore intrinsically limited. The major challenges in track mode are estimating accurate star Centroid even with high electronic noise in static and dynamic conditions, increasing angular rate, selection of correct star for attitude determination and better through put for control application. We are presenting design of state-of-art track mode software modules to address all these challenges and overcome the limitations of conventional star trackers.

Mark-II track mode operation starts with predicting probable star positions in the FOV based on the previously computed attitude, rate and acceleration. These positions are used to define active windows in the FOV. Detector is exposed to a well defined time called integration time and detector is drive to process only these selected areas and digitized pixels information is available in the shared memory. The operations performed on these pixels data for effective attitude and rate estimation is shown in the below pseudo code.

Pseudo-code Track Mode Operations in Mark-II Star sensor

1: **for** M = 1 to M = Number of Track windows **do**
2: Estimate the variable background for each window.
3: Compute the centre of star for each window.
4: Correct Distortion for each star.
5: Check for Bright Object in window.
6: Check for bad pixels in window.
5: **end for**
6: Verify the position of each measured star with its predicted star position.
7: Verify the magnitude coherency of each measured star With reference magnitude in Onboard Star Catalog.
8: Select valid stars in all aspect for attitude estimation.
9: Compute the attitude for the current exposure.
10: Estimate the satellite rate and acceleration using latest 3 Attitude samples.
11: Format the data for telemetry and send to AOCS.

In Mark-II star sensor, sufficient logics are built to track stars even at high angular rates. This is accomplished through increasing the size of the track window and through two novel concepts namely auto-integration and binning. In all cases of tracking, various coherency checks are carried out to select only the valid stars for attitude and rate estimation. Normally all the above tracking operations are carried out in a sequential fashion, starting from integrating detector, processing pixels to computing attitude. This mode of track is called 'Disjoint' Track Mode and throughput is limited to 4 Hz.

In order to improve the throughput, massive software parallelism is implemented, wherein when integration of current exposure is in progress, the processor process the previous exposure data. This pipelining parallelism yields 8 Hz throughput as shown in Fig.3.

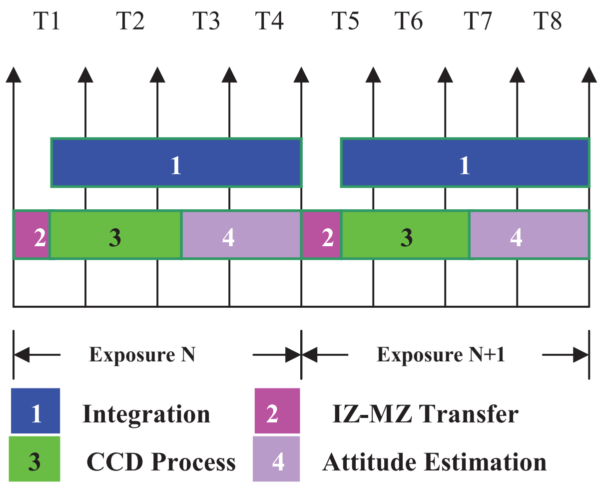

Fig.3: Timing of Parallel Mode Operations.

As shown in the Fig.3, T1, T2..Tn represents the Time Reference (TREF) pulse from AOCS, the duration of each pulse is 32 msec. All star tracker activities are synchronized with respect to this pulse. Each exposure will take 4 TREF pulses, and when integration of current exposure is in progress (Activity-1), previous exposure data will be processed in parallel with integration. These activities are transferring Image zone of detector to Memory zone (Activity-2), CCD readout (Activity-3) and Estimation of attitude and formatting data for AOCS (Activity-4). Since each exposure will take only 4 TREF pulses, 8Hz update rate is achieved. If the above activities are carried out sequentially we can achieve only 4 Hz update rate, which is not adequate for proper controlling applications. The software parallelism incorporated in the star tracker software enables to provide higher update rate even with the low speed hardware, which is mainly due to the constraints of harsh space environment.

VII. SIMULATION AND TESTING

Testing of algorithms is really a challenging task, because the algorithm has to test for different real time space conditions by simulating the suitable test patterns. The algorithm is implemented on ERC32 target computer and tested for different cases by generating test pattern using a highly sophisticated test facility called Dynamic Multi Star Simulator (DMSS), which simulates real sky scenario corresponding to different orientations in both Initial and Final Bench Test. The display logic in DMSS is based on X window system (Xlib programming written in C) [4]. The following are the different test conducted on target with DMSS in addition to these tests flight acceptance tests are also successfully completed.

1. Random Pattern Test: Star patterns are generated randomly for different right ascension and declination. The validity of the LIS software modules are ensured by generating more than 10000 patterns. The summary of LIS test result is shown in Table-2.
2. Orbit Test: Star patterns are different for each orbits, these patterns are generated based on some orbital parameters with nominal spacecraft speed. The track mode algorithms are tested in this mode, the performance is shown in Fig.4.
3. Rate Test: To ensure successful LIS and subsequently track even at higher angular.

TABLE-1

SUMMARY OF IDENTIFICATION TEST RESULTS

Sl No	Test Type	Algorithm- A Success Rate	Algorithm- B Success Rate
1	Random Pattern Test	90	99
2	Orbit test	90	98
3	Globe Coverage Test	90	98
4	Rate Test - 1.0°/sec	30	90
5	Rate Test – 2.0°/sec	10	84

Fig.4: Tracking performance in front of DMSS

As shown in Fig.4, the star tracker is always tracking 9 to 10 stars in the FOV and almost same numbers of stars are used for estimating attitude. To perform all operations the tracker consuming less than 128 msec and thus provides 8 Hz update for controlling. The software presented here is embedded in Mark-II star sensor which is flown in Indian prestigious SARAL mission as a prime attitude sensor. After power ON within few secs sensor provides the LIS solution and subsequently starts tracking 10 stars and estimating both attitude and rate satisfactorily. The performance of sensor is excellent in all types of satellite operations. The noise performance of attitude estimated by star tracker is well within specification, as shown in Fig.5 the boresight noise is about 6 arc sec and cross axis noise is about 3 arc sec. All sensor parameters are within specification and all software logics are evaluated to be normal and functioning as designed.

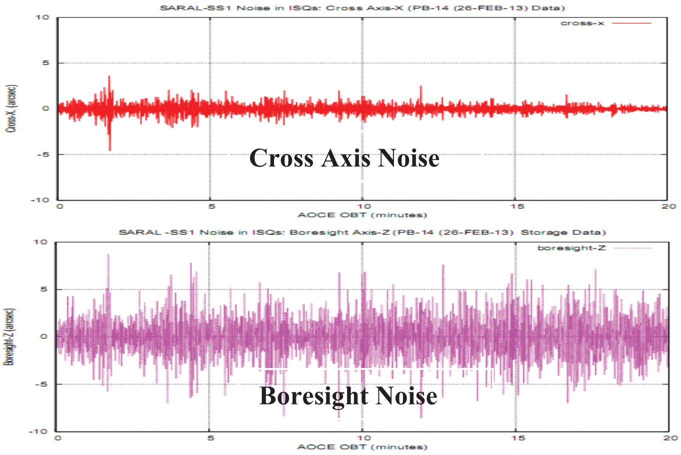

Fig.5: Onboard Cross and Boresight noise.

IX. CONCLUSIONS

Innovative software used in Mark-II sensor to provide faster Acquisition and track mode attitude for Mark-II star tracker is presented here. The software is tested at different stages using highly sophisticated simulators and performance is normal. The star tracker with this software is flown in Indian SARAL mission and post performance of tracker is excellent. The tracker provides attitude with required accuracy with specified update rate. All the software logics built in the system are functioning normally in all space conditions and different satellite operations and make the sensor work-horse for future ISRO programmes ranging from remote sensing applications to inter planetary missions, which includes missions like Navigation satellites, Mangalyaan and Chandrayaan-2.

X. ACKNOWLEDGEMENTS

The authors wish to express their sincere thanks to Shri J.A Kamalakar, Director, LEOS for encouragement and support.

XI. REFERENCES

[1] Carl Christian Liebe, *Accuracy Performance Performance of Star Trackers – A Tutorial*, IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS VOL. 38, NO. 2 APRIL 2002.

[2] Mortari, D.; Samaan, M. A.; Bruccoleri, C. The pyramid star identification technique. Navigation. 2004, 51, 171–183.

[3] Rafael Gonzalez , Richard E Woods, “Digital Image Processing”, Pearson Education

[4] James Gettys, “Xlib- C language, X Interface”, X Consortium Standard, X Version 11 Release 6.7 Draft.